

# JAVA.MATH.BIGINTEGER CLASS

[http://www.tutorialspoint.com/java/math/java\\_math\\_biginteger.htm](http://www.tutorialspoint.com/java/math/java_math_biginteger.htm)

Copyright © tutorialspoint.com

## Introduction

The **java.math.BigInteger** class provides operations analogues to all of Java's primitive integer operators and for all relevant methods from `java.lang.Math`.

It also provides operations for modular arithmetic, GCD calculation, primality testing, prime generation, bit manipulation, and a few other miscellaneous operations. All operations behave as if `BigInteger`s were represented in two's-complement notation.

Semantics of arithmetic operations and bitwise logical operations are similar to those of Java's integer arithmetic operators and Java's bitwise integer operators respectively. Semantics of shift operations extend those of Java's shift operators to allow for negative shift distances.

Comparison operations perform signed integer comparisons. Modular arithmetic operations are provided to compute residues, perform exponentiation, and compute multiplicative inverses. Bit operations operate on a single bit of the two's-complement representation of their operand.

All methods and constructors in this class throw `NullPointerException` when passed a null object reference for any input parameter.

## Class declaration

Following is the declaration for **java.math.BigInteger** class:

```
public class BigInteger
    extends Number
        implements Comparable<BigInteger>
```

## Field

Following are the fields for **java.math.BigInteger** class:

- **static BigInteger ONE** -- The `BigInteger` constant one.
- **static BigInteger TEN** -- The `BigInteger` constant ten.
- **static BigInteger ZERO** -- The `BigInteger` constant zero.

## Class constructors

S.N.	Constructor & Description
1	<b>BigInteger(byte[] val)</b> This constructor is used to translate a byte array containing the two's-complement binary representation of a <code>BigInteger</code> into a <code>BigInteger</code> .
2	<b>BigInteger(int signum, byte[] magnitude)</b> This constructor is used to translate the sign-magnitude representation of a <code>BigInteger</code> into a <code>BigInteger</code> .
3	<b>BigInteger(int bitLength, int certainty, Random rnd)</b>

	This constructor is used to construct a randomly generated positive BigInteger that is probably prime, with the specified bitLength.
4	<b>BigInteger(int numBits, Random rnd)</b> This constructor is used to construct a randomly generated BigInteger, uniformly distributed over the range 0 to $(2^{\text{numBits}} - 1)$ , inclusive.
5	<b>BigInteger(String val)</b> This constructor is used to translate the decimal String representation of a BigInteger into a BigInteger.
6	<b>BigInteger(String val, int radix)</b> This constructor is used to translate the String representation of a BigInteger in the specified radix into a BigInteger.

## Class methods

S.N.	Method & Description
1	<a href="#"><u>BigInteger abs()</u></a> This method returns a BigInteger whose value is the absolute value of this BigInteger.
2	<a href="#"><u>BigInteger add(BigInteger val)</u></a> This method returns a BigInteger whose value is (this + val).
3	<a href="#"><u>BigInteger and(BigInteger val)</u></a> This method returns a BigInteger whose value is (this & val).
4	<a href="#"><u>BigInteger andNot(BigInteger val)</u></a> This method returns a BigInteger whose value is (this & ~val).
5	<a href="#"><u>int bitCount()</u></a> This method returns the number of bits in the two's complement representation of this BigInteger that differ from its sign bit.
6	<a href="#"><u>int bitLength()</u></a> This method returns the number of bits in the minimal two's-complement representation of this BigInteger, excluding a sign bit.
7	<a href="#"><u>BigInteger clearBit(int n)</u></a> This method returns a BigInteger whose value is equivalent to this BigInteger with the designated bit cleared.
8	<a href="#"><u>int compareTo(BigInteger val)</u></a> This method compares this BigInteger with the specified BigInteger.
9	<a href="#"><u>BigInteger divide(BigInteger val)</u></a> This method returns a BigInteger whose value is (this / val).
10	<a href="#"><u>BigInteger[] divideAndRemainder(BigInteger val)</u></a> This method returns an array of two BigIntegers containing (this / val) followed by (this % val).
11	<a href="#"><u>double doubleValue()</u></a> This method converts this BigInteger to a double.
12	<a href="#"><u>boolean equals(Object x)</u></a>

	This method compares this BigInteger with the specified Object for equality.
13	<a href="#"><u>BigInteger flipBit(int n)</u></a> This method returns a BigInteger whose value is equivalent to this BigInteger with the designated bit flipped.
14	<a href="#"><u>float floatValue()</u></a> This method converts this BigInteger to a float.
15	<a href="#"><u>BigInteger gcd(BigInteger val)</u></a> This method returns a BigInteger whose value is the greatest common divisor of abs(this) and abs(val).
16	<a href="#"><u>int getLowestSetBit()</u></a> This method returns the index of the rightmost (lowest-order) one bit in this BigInteger (the number of zero bits to the right of the rightmost one bit).
17	<a href="#"><u>int hashCode()</u></a> This method returns the hash code for this BigInteger.
18	<a href="#"><u>int intValue()</u></a> This method converts this BigInteger to an int.
19	<a href="#"><u>boolean isProbablePrime(int certainty)</u></a> This method returns true if this BigInteger is probably prime, false if it's definitely composite.
20	<a href="#"><u>long longValue()</u></a> This method converts this BigInteger to a long.
21	<a href="#"><u>BigInteger max(BigInteger val)</u></a> This method returns the maximum of this BigInteger and val.
22	<a href="#"><u>BigInteger min(BigInteger val)</u></a> This method returns the minimum of this BigInteger and val.
23	<a href="#"><u>BigInteger mod(BigInteger m)</u></a> This method returns a BigInteger whose value is (this mod m).
24	<a href="#"><u>BigInteger modInverse(BigInteger m)</u></a> This method returns a BigInteger whose value is (this <sup>-1</sup> mod m).
25	<a href="#"><u>BigInteger modPow(BigInteger exponent, BigInteger m)</u></a> This method returns a BigInteger whose value is (this <sup>exponent</sup> mod m).
26	<a href="#"><u>BigInteger multiply(BigInteger val)</u></a> This method returns a BigInteger whose value is (this * val).
27	<a href="#"><u>BigInteger negate()</u></a> This method returns a BigInteger whose value is (-this).
28	<a href="#"><u>BigInteger nextProbablePrime()</u></a> This method returns the first integer greater than this BigInteger that is probably prime.
29	<a href="#"><u>BigInteger not()</u></a> This method returns a BigInteger whose value is (~this).
30	<a href="#"><u>BigInteger or(BigInteger val)</u></a> This method returns a BigInteger whose value is (this   val).

31	<a href="#"><u>BigInteger pow(int exponent)</u></a> This method returns a BigInteger whose value is (this <sup>exponent</sup> ).
32	<a href="#"><u>static BigInteger probablePrime(int bitLength, Random rnd)</u></a> This method returns a positive BigInteger that is probably prime, with the specified bitLength.
33	<a href="#"><u>BigInteger remainder(BigInteger val)</u></a> This method returns a BigInteger whose value is (this % val).
34	<a href="#"><u>BigInteger setBit(int n)</u></a> This method returns a BigInteger whose value is equivalent to this BigInteger with the designated bit set.
35	<a href="#"><u>BigInteger shiftLeft(int n)</u></a> This method returns a BigInteger whose value is (this << n).
36	<a href="#"><u>BigInteger shiftRight(int n)</u></a> This method returns a BigInteger whose value is (this >> n).
37	<a href="#"><u>int signum()</u></a> This method returns the signum function of this BigInteger.
38	<a href="#"><u>BigInteger subtract(BigInteger val)</u></a> This method returns a BigInteger whose value is (this - val).
39	<a href="#"><u>boolean testBit(int n)</u></a> This method returns true if and only if the designated bit is set.
40	<a href="#"><u>byte[] toByteArray()</u></a> This method returns a byte array containing the two's-complement representation of this BigInteger.
41	<a href="#"><u>String toString()</u></a> This method returns the decimal String representation of this BigInteger.
42	<a href="#"><u>String toString(int radix)</u></a> This method returns the String representation of this BigInteger in the given radix.
43	<a href="#"><u>static BigInteger valueOf(long val)</u></a> This method returns a BigInteger whose value is equal to that of the specified long.
44	<a href="#"><u>BigInteger xor(BigInteger val)</u></a> This method returns a BigInteger whose value is (this ^ val).